ET420197679US

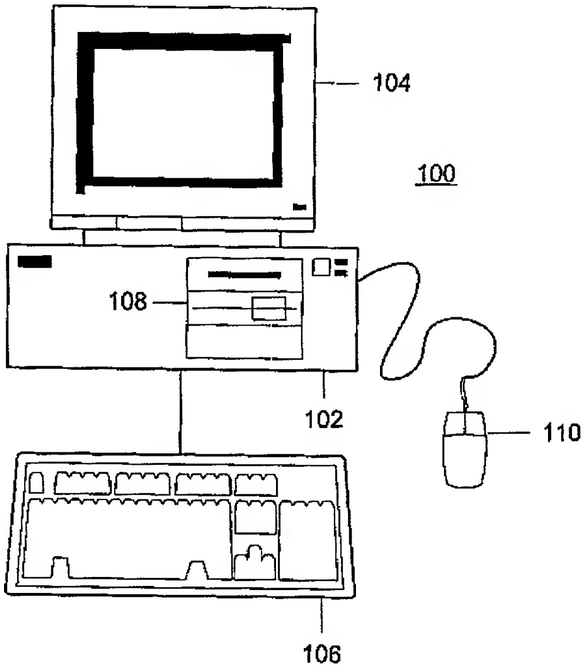


Figure 1

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including
Functions and Equalities
Page 1 of 10

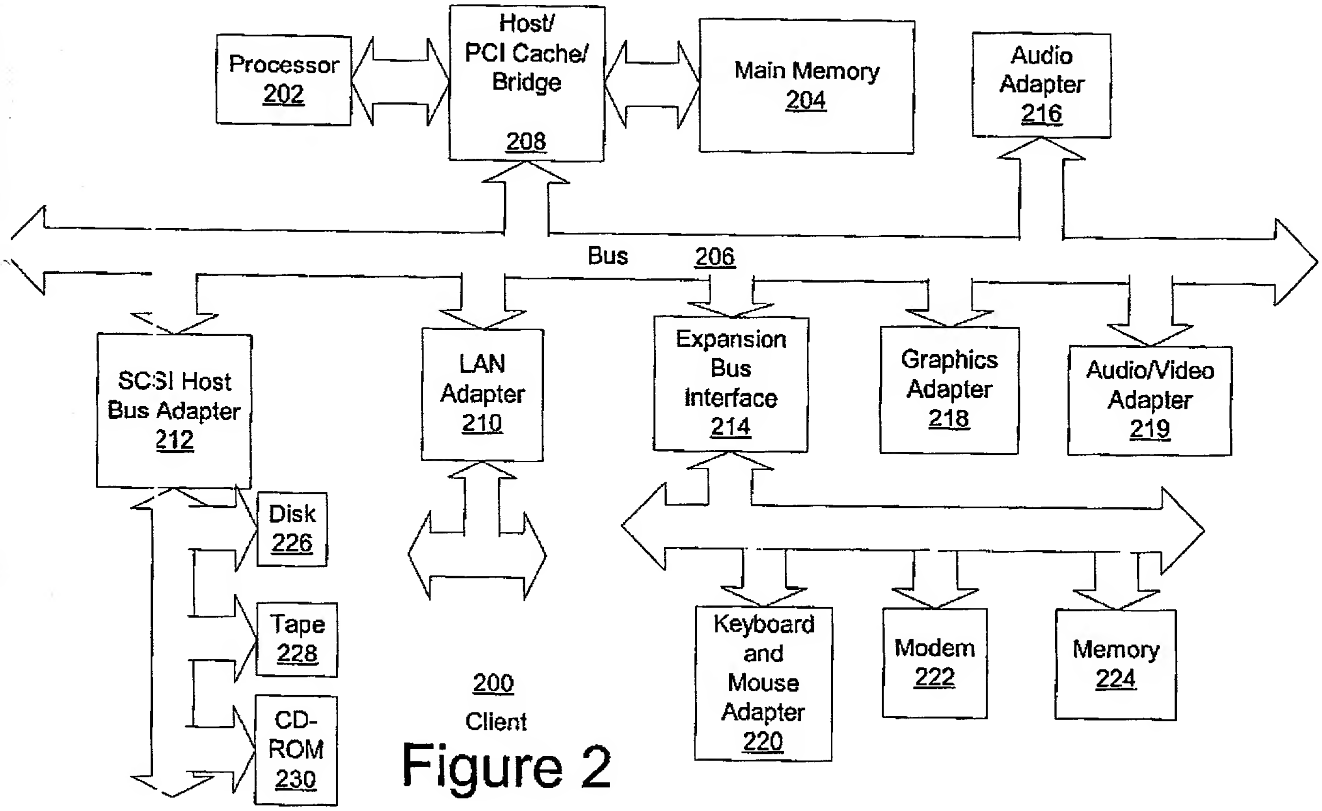
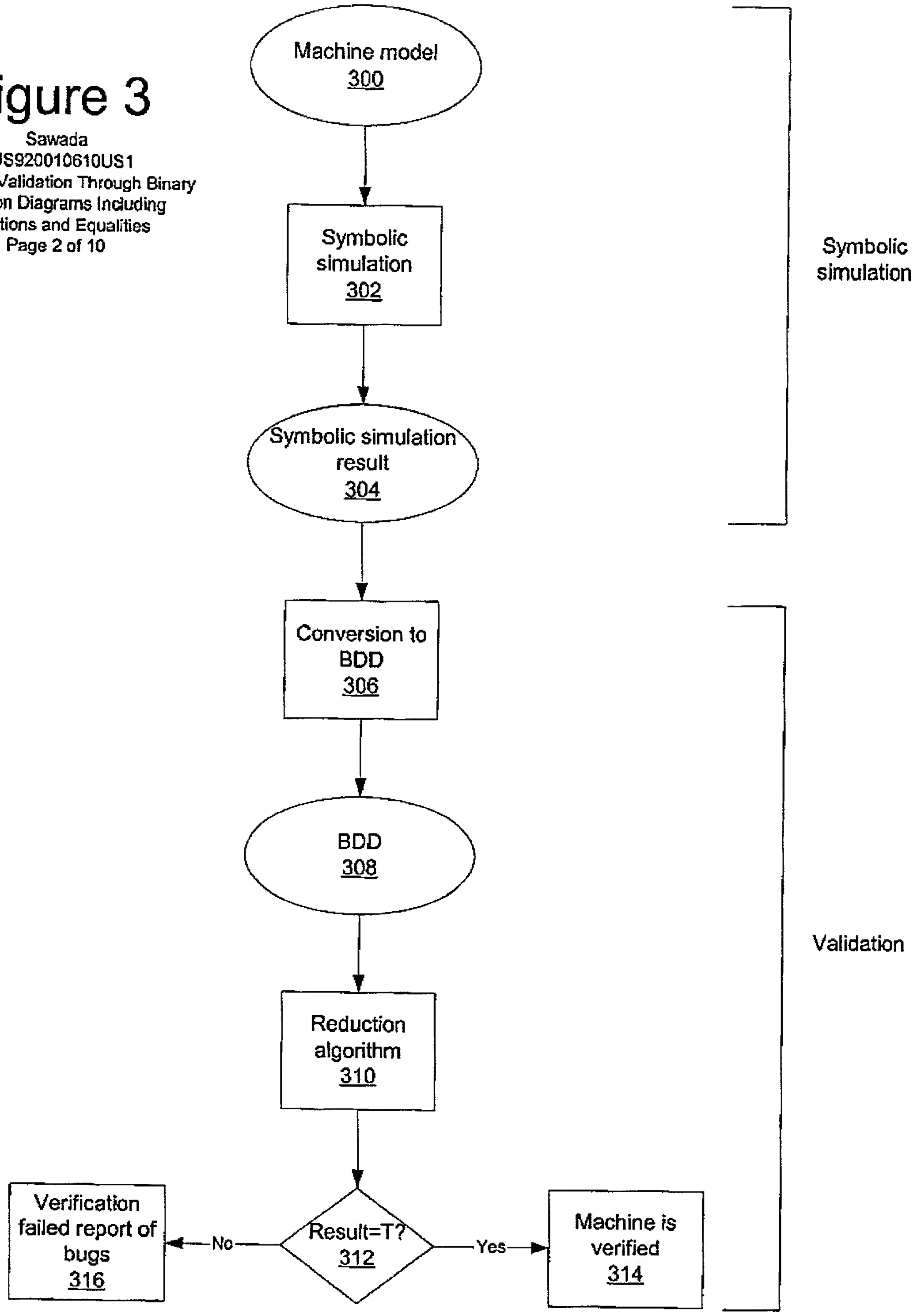


Figure 2

Figure 3

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including
Functions and Equalities
Page 2 of 10



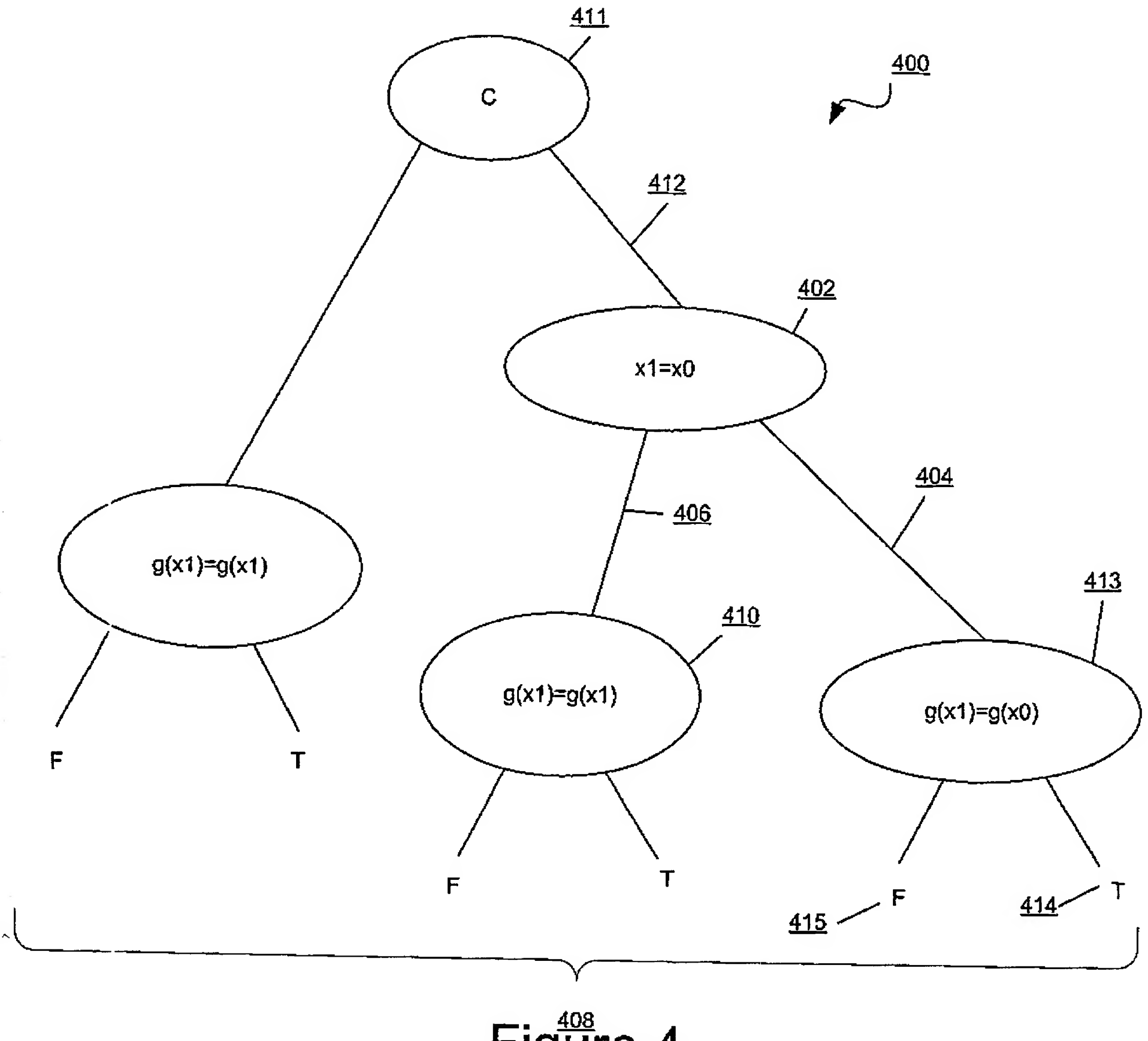


Figure 4

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including Functions
and Equalities

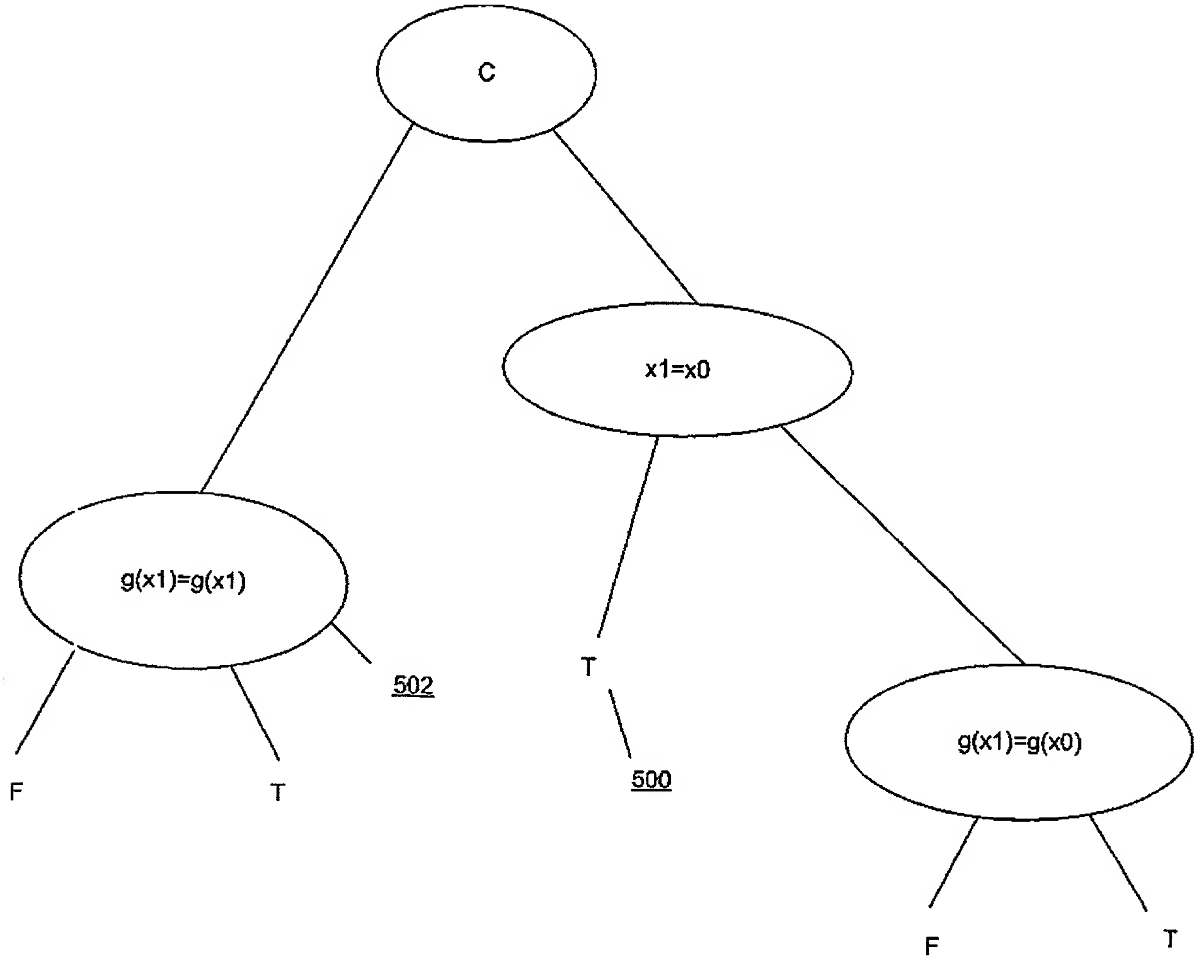


Figure 5

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including Functions
and Equalities

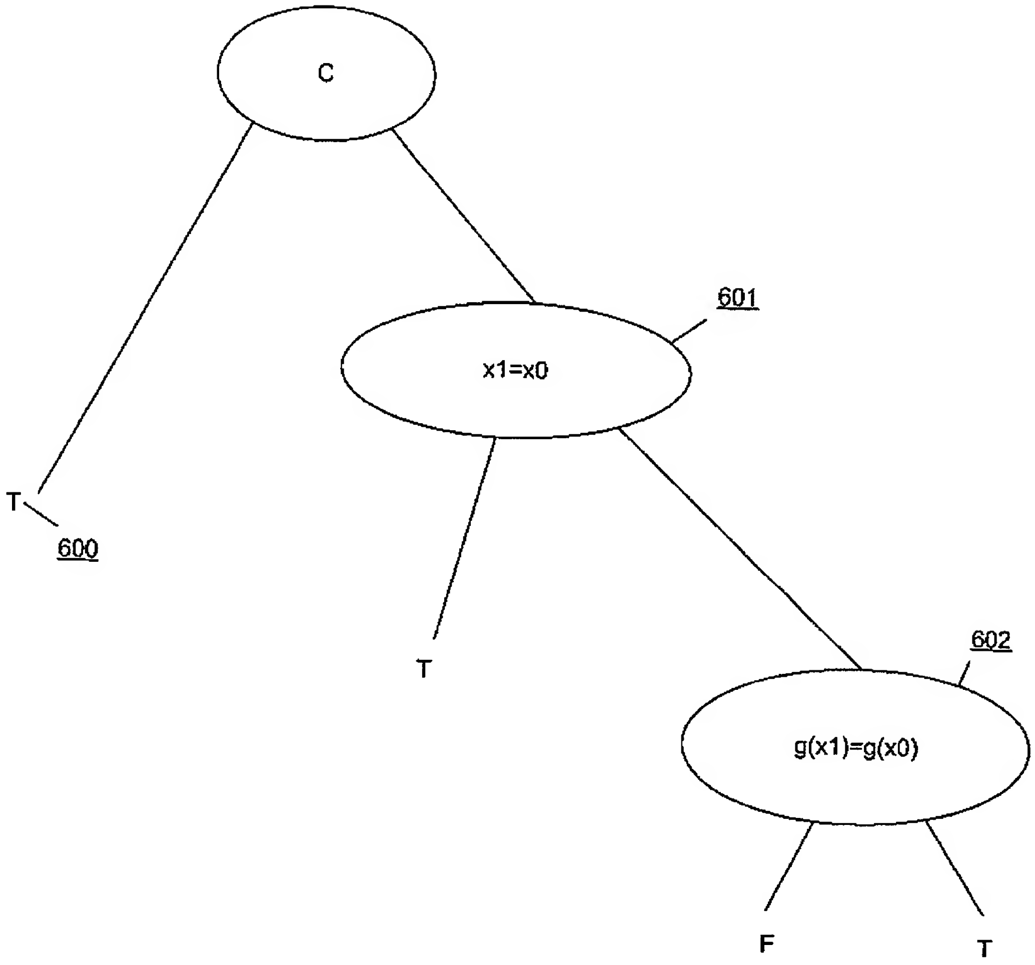


Figure 6

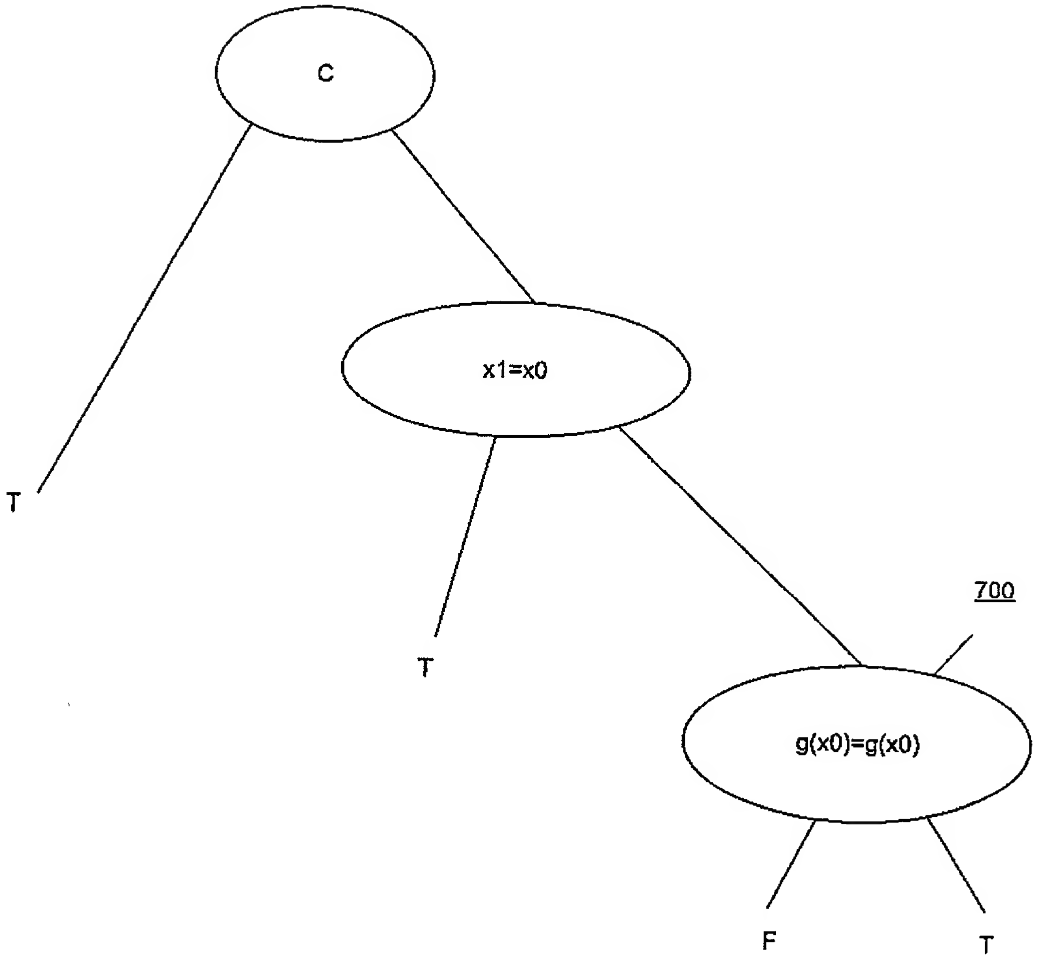


Figure 7

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including Functions
and Equalities

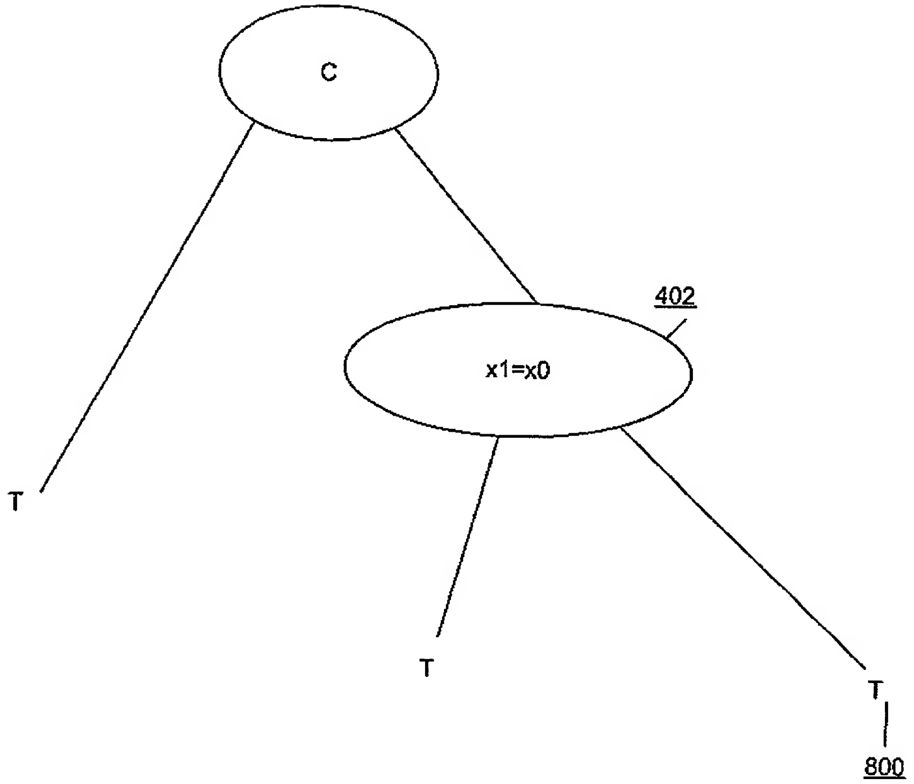


Figure 8

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including Functions
and Equalities

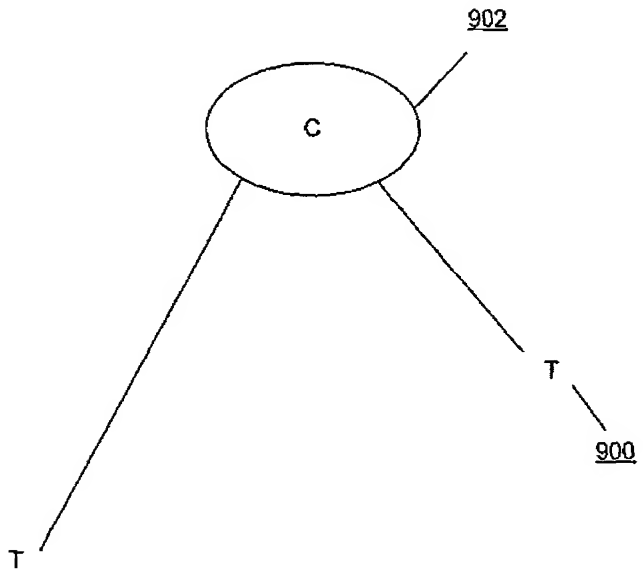


Figure 9

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including Functions
and Equalities
Page 8 of 10

Figure 10

1100

```
1101  simplify (X,Y) :- sim (X,Z), !, simplify (Z,Y).
      simplify (X,X).

      sim(ite(S=S, H, _), H).
      sim(ite(S=T,H,K), ite(T=S,H,K)) :- gt(T,S).
      sim(ite(_,H,H), H).
      sim(ite(E, ite(E,H,_), L), ite(E,H,L)).
      sim(ite(E, H, ite(E,_,L)), ite(E,H,L)).
1102  sim(ite(E1, ite(E2,H,K), L), ite(E2, ite(E1,H,L), ite(E1,K,L))) :-
            gts(E1,E2).
      sim(ite(E1, H, ite(E2,K,L)), ite(E2, ite(E1,H,K), ite(E1,H,L))) :-
            gts(E1,E2).
      sim(ite(S=T, H, K), ite(S=T, L, K)) :-
            gt(S,T),repl(S,T,H,L), H \==L.          1104
1108  sim(ite(A,B,C), ite(A,X,C)) :- sim(B,X).
      sim(ite(A,B,C), ite(A,B,X)) :- sim(C,X).

      repl(S,T,S,T):- !.
      repl(S,T,P,Q) :-
             P = .. [X|Args],
6            maprepl(S, T, Args, Newargs),
             Q = .. [X|Newargs].

      naprepl(_,_,[],[]).
      naprepl(S, T, [X|A], [Y|B]) :- repl(S,T,X,Y), maprepl (S,T,A,B).

10    gts(A=_, C=D) :- gt(A,C), gt(A,D).
      gts(_=B, C=D) :- gt(B,C), gt(B,D).

2     depth(P,X) :- atom(P), !, X is 0.
      depth(P,X) :- P =.. [_|Args], max_depth(Args,Y), X is Y+1.

      nax_depth([],0).
      nax_depth([A|L],X) :- depth(A,Y), max_depth(L,Z), X is max(Y,Z).

      gt(P,Q) :- depth(P,DP), depth(Q,DQ), DP>DQ, !.
      gt(P,Q) :- P =.. [F|_], Q =.. [G|_], F \== G, gtlex(F,G), !.
      gt(P,Q) :- P=..[F|Args1], Q =.. [F|Args2], gtlist(Args1,Args2).

      gtlist([A1|L1],[A2|L2]) :- A1\==A2, !, gt(A1,A2).
      gtlist([_|L1],[_|L2]) :- gtlist(L1,L2).

1114  gtlex(g,f).
      gtlex(x1,x0).
```

Figure 11

Sawada
AUS920010610US1
Hardware Validation Through Binary
Decision Diagrams Including
Functions and Equalities

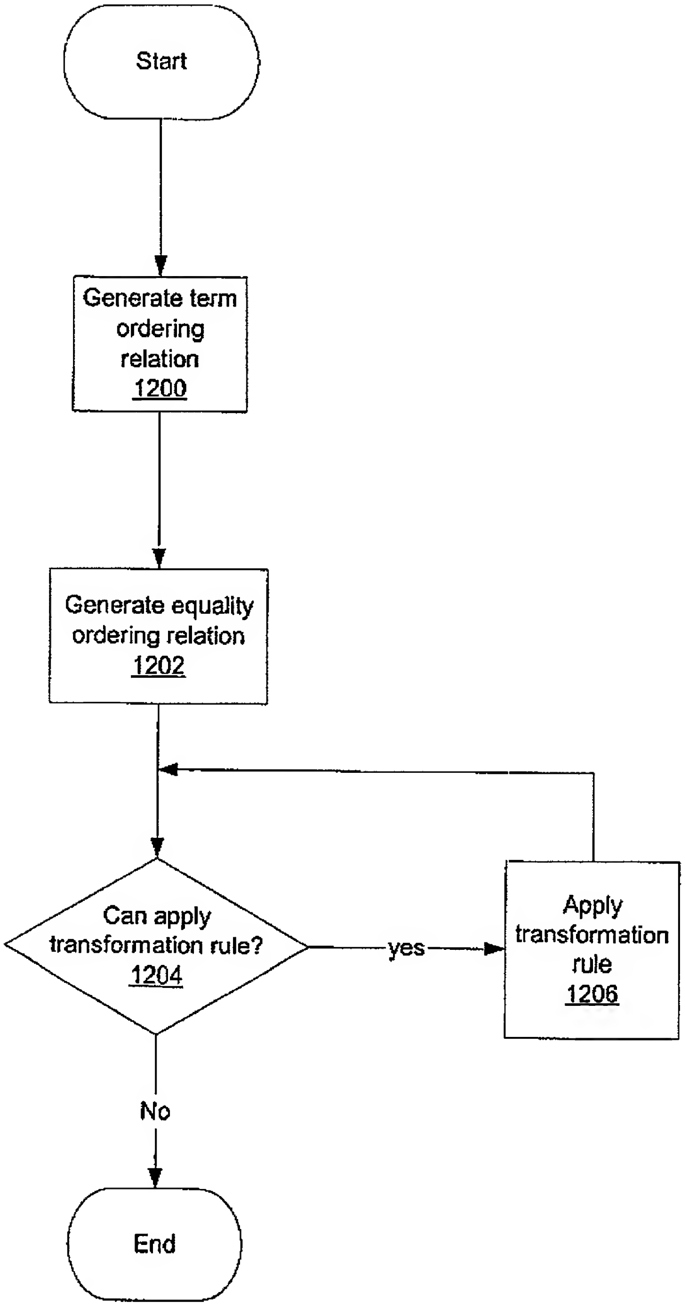

Start
Generate term ordering relation 1200
Generate equality ordering relation 1202
Can apply transformation rule? 1204
yes
Apply transformation rule 1206
No
End

Figure 12

Sawada
AUS920010610US1
Hardware Validation Through Binary Decision Diagrams Including Functions and Equalities